

Advanced Software Development Workstation

OOPSLA 92 Conference

Michel Izygon

Barrios Technology, Inc.

December 31, 1992

*GRANT
IN-60-CR
157386
P. 7*

N93-22041

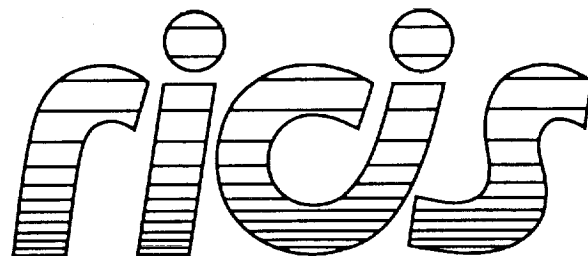
Unclas

G3/60 0157386

**Cooperative Agreement NCC 9-16
Research Activity No. SR.02**

**NASA Johnson Space Center
Information Systems Directorate
Technology Development Division**

(NASA-CR-192811) ADVANCED SOFTWARE
DEVELOPMENT WORKSTATION. OOPSLA
1992 CONFERENCE. TRIP REPORT
(Research Inst. for Computing and
Information Systems) 7 p



*Research Institute for Computing and Information Systems
University of Houston-Clear Lake*

TRIP REPORT

The RICIS Concept

The University of Houston-Clear Lake established the Research Institute for Computing and Information Systems (RICIS) in 1986 to encourage the NASA Johnson Space Center (JSC) and local industry to actively support research in the computing and information sciences. As part of this endeavor, UHCL proposed a partnership with JSC to jointly define and manage an integrated program of research in advanced data processing technology needed for JSC's main missions, including administrative, engineering and science responsibilities. JSC agreed and entered into a continuing cooperative agreement with UHCL beginning in May 1986, to jointly plan and execute such research through RICIS. Additionally, under Cooperative Agreement NCC 9-16, computing and educational facilities are shared by the two institutions to conduct the research.

The UHCL/RICIS mission is to conduct, coordinate, and disseminate research and professional level education in computing and information systems to serve the needs of the government, industry, community and academia. RICIS combines resources of UHCL and its gateway affiliates to research and develop materials, prototypes and publications on topics of mutual interest to its sponsors and researchers. Within UHCL, the mission is being implemented through interdisciplinary involvement of faculty and students from each of the four schools: Business and Public Administration, Education, Human Sciences and Humanities, and Natural and Applied Sciences. RICIS also collaborates with industry in a companion program. This program is focused on serving the research and advanced development needs of industry.

Moreover, UHCL established relationships with other universities and research organizations, having common research interests, to provide additional sources of expertise to conduct needed research. For example, UHCL has entered into a special partnership with Texas A&M University to help oversee RICIS research and education programs, while other research organizations are involved via the "gateway" concept.

A major role of RICIS then is to find the best match of sponsors, researchers and research objectives to advance knowledge in the computing and information sciences. RICIS, working jointly with its sponsors, advises on research needs, recommends principals for conducting the research, provides technical and administrative support to coordinate the research and integrates technical results into the goals of UHCL, NASA/JSC and industry.

RICIS Preface

This research was conducted under auspices of the Research Institute for Computing and Information Systems by Dr. Michel Izygon of Barrios Technology, Inc. Dr. Rodney L. Bown served as the RICIS research coordinator.

Funding was provided by the Information Systems Directorate, NASA/JSC through Cooperative Agreement NCC 9-16 between the NASA Johnson Space Center and the University of Houston-Clear Lake. The NASA research coordinator for this activity was Ernest M. Fridge III, Deputy Chief of the Software Technology Branch, Technology Development Division, Information Systems Directorate, NASA/JSC.

The views and conclusions contained in this report are those of the author and should not be interpreted as representative of the official policies, either express or implied, of UHCL, RICIS, NASA or the United States Government.

Advanced Software Development Workstation

OOPSLA 92 Conference Trip Report

**Prepared for
NASA-Johnson Space Center**

December 31, 1992

**Submitted by
Dr. Michel Izygon
Barrios Technology Inc.
1331 Gemini Av.
Houston, TEXAS 77058**

ABSTRACT

This report summarizes the main trends observed at the Object Oriented Programming Systems, Languages, and Applications Conference held in Vancouver, British Columbia. The conference was held from October 19 to October 22, 1992. This Conference is the main Object-Oriented Event that allows us to assess the dynamism of the technology and to meet the main actors of the field. It is an invaluable source of information for the ASDW project.

OOPSLA Conference Report
RICIS Project
Michel Izygon

Summary:

More and more organizations are moving to Object-Oriented Software Development. The Software Development Process is being adapted to take full advantage of the OO paradigm. Object-Oriented methods are flourishing and are now reaching their maturity level. Tools are being developed to provide more support to these methods. The language that stands out from the conference is C++. It is used in about 90% of the projects.

Main trends:

We want to summarize here the main trends detected during OOPSLA in the following areas: Object Technology usage, OO methods, OO Tools, OO Languages.

- General Trend in Object Technology:

The main trend shown in the conference is that more and more organizations are moving to Object-Oriented Technology. A sign of this trend can be given by the number of participants to the conference: About 2500 people attended the 92 conference as compared to 1500 last year. Another sign of this trend is given by the growing number of experience reports for which three full sessions were devoted. Noteworthy mentioning, most reports were about success stories. In some plenary sessions held by long time Object Gurus, such as Grady Booch, the message was that it's not a matter of if companies should move to this new paradigm, but when and how they should do it. The Object paradigm is referred as the ultimate, final technology in Software Engineering. It is viewed as the "end of the software engineering history." Though, there were also arguments cautioning that this technology, cannot solve by itself the software crisis. "Technologies are not a substitute for good processes"; efficient processes are viewed by some speakers as the real solution, when based on a good technology. Much effort is consequently being put into adapting the Software Development Process to the Object paradigm.

The MIS world is felt as being the forum where success or failure of the technology will be assessed. Managers seem to be aware of the potential benefits they can get from Object technology, but they don't want to move too early, as some fears were echoed that OO could be the AI of the 90s (AI taken as the example of technology that failed in the 80s). The field is currently in a very dynamic state. Books containing the term Object-Oriented in their title are flourishing at a high rate, and one must admit that some of them contain very useful pieces of information to help a software engineer or an organization adopting the new paradigm. The number of consultants in the field is also growing rapidly. Everyone knowing some Object concept starts a company to offer services in training or in helping organizations to use the technology. No doubt that a consolidation is needed and will probably happen in the next few years, when every company will have its own Object group. This has certainly started. Representative of many Fortune 100 companies were present at the conference as observers as well as participants. It seems that many of these companies have their OO group working at starting some projects using the technology, and being seen as change actors within the organization.

- Trend in Object-Oriented Methodologies:

This OOPSLA conference has seen a multiplication of OO methods. After the 90-91 very rich harvest that included Booch, Schlaer Mellor, Rumbaugh, Coad Yourdon, Wirfs-

Brock, one would have thought that opportunities for new methods were fading. Instead, the OO Analysis and Design methods have proliferated at an even higher rate. These include James Martin, Embley, and Jacobson. The official way to create a method being the publication of a book, we can mention also the book written by Berard, a long time OO expert that has developed a method taught for the past few years in his training classes but not published up to now. Two more books worth mentioning, because they will probably have a significant impact on the OO community, will appear in the next three months. These define two new methods: the first one is named the Fusion method, the goal of which is to use the best features of Rumbaugh, Booch and the CRC cards concept; the second is the method developed by Dennis de Champeaux from HP Labs, emphasizing the development principles of OO software. No doubt that many more books will be published during the coming year. There is a consensus that today's methods are first generation methods that will soon be followed by second generation methods that will correct many weaknesses of the first ones by complementing them and making the transition between Analysis, Design and Coding easier. These second generation methods may include the two books above mentioned and the second editions of the Booch and Rumbaugh's books that are supposed to come by the end of 93. In fact Booch will come out with a second book by the end of 93, apart from the new edition of his previous one, that will concentrate on the software development process. Worth noticing are the attempts by Booch to have Rumbaugh and him working together on a common method. These attempts did not seem to interest particularly Rumbaugh. As far as method usage is concerned, it seems clear that the two leaders are Booch and Rumbaugh. Another noticeable trend that I already mentioned in the previous section is the focus on the software development process, viewed now as one key element of any successful project. More and more efforts are therefore done on the subject in order to develop the appropriate process adapted to the OO paradigm and integrated with the OOAD method. The Booch's book and the de Champeaux's book referred to previously, are just two examples of this trend. Adele Goldberg of ParcPlace also presented in detail her view on the subject during some Sessions. The OO software development process was at the center of one Panel Session and one Workshop during the conference.

- Trend in Tools and Environments

This year has seen a big increase in the number of tools and development environments available. As far as Object Oriented tools are concerned, the main trend is the appearance on the market of tools supporting OOA and/or OOD methods. Almost each well-known method has a corresponding tool, often developed by a company that has some links to the method's developer. Booch's method is directly supported by a product named ROSE, developed by Rational for which Grady Booch is working. Rumbaugh's method is supported by OMTool developed by General Electric Advanced Concept Center, while Jim Rumbaugh is working at GE Corporate Research & Development. Coad Yourdon's method is supported by OOATool, developed by Object International whose Chairman is Peter Coad himself. Jacobson's method is supported by a tool named Objectory (Also the name of the method) developed by Objective Systems whose Chairman is Ivar Jacobson. James Martin's method is supported by a tool developed by Intellicorp, after this company has been bought by James Martin. Other mainstream SA/SD type of tools such as Cadre Teamwork, IDE STP, are adding OO features, the first one to support the Schlaer Mellor method, the second one supporting OOSD, CEO Wasserman's own method. All the previously mentioned tools are one-method-oriented, i.e., they support only one method. If the methods can be considered as first generation methods, the tools are even more clearly suffering from some youth weaknesses. For most of them, they support the method only partially. More of a problem is the fact that, for an organization, buying a tool means to adopt the method it supports, and it eventually may mean being stuck with it, even if other methods are prevailing in the near future. Multiple method tools are

now also available. Apart from Paradigm+, there are Object Maker from Mark V Systems and Virtual Software Factory from VSFL that implement different methods and allow one to implement his own method. From the demonstrations given during OOPSLA, the most mature of these tools is without doubt Paradigm+. It is clearly superior in functionality and more robust. This type of tool is what we believe has the best potential for any organization beginning to use the Object paradigm, without worrying on what will be the standard OOAD method in two or three years. The Paradigm+ implementation of Rumbaugh's method is also clearly the best one available today. Noteworthy mentioning is the fact that it will support the Fusion method as soon as the book is published.

As far as development environments are concerned, support to C++ development reached the same level of maturity as for C.

- Trend in Languages:

The language that appears to be largely dominant on the Object-Oriented Programming languages scene is C++. Ada was totally absent from the conference (this has sparked some reactions from the Ada community on the InterNet News, and probably next year Ada will have a higher profile). Smalltalk has still a tiny portion of the market (about 10-20%) for its prototyping capabilities. Eiffel is still present but more as a training tool to teach the Object Oriented technology than to develop real world projects. C++ seems to have won the battle, regardless of all its defects. A major sign of this trend is the rapid expansion of the C++ development tools offered. Object Center from CenterLine, ObjectWorks for C++ are two of the main new and impressive development environments available that will allow probably many C developers to move advantageously to C++. Another sign of the trend is given by the fact that Booch, early Ada believer, has moved on to C++ and has favorably compared C++ to Ada. The advent of the Object Oriented Paradigm is pushing Ada out of the picture even though it is recognized to be probably easier to maintain. The power of the use of a real OO language is perceived as more beneficial than the good software engineering concepts embedded into Ada. The coming of Ada 9X might change the picture, unless it comes too late. The two languages are perceived to be on a collision course as far as their features are concerned.

